521282S Biosignal processing II (Lab exercises, spring 2019)

Lab – I Artifact removal from EEG

Objective

The aim of this exercise is to learn how to filter electroencephalogram (EEG) signals containing noise (e.g. power line interference (PLI), white noise, and unwanted trends) and unwanted artifacts (e.g. eye movements and blinks, heart signals, muscle artifacts and other artifacts residing in the EEG bands of interest).

A sample of 60 second real EEG data of an awake person with 13 channels + 2 electrooculogram (EOG) reference channels containing PLI noise, random trends, and other artifacts is used.

In the exercise, first, basic filters are designed to rid of the common frequency artifacts (i.e. PLI and trends). Next, adaptive filtering is applied to demonstrate EOG filtering using the two reference EOG channels. Finally, an alternative approach using a blind source separation (BSS) technique to find and remove EOG and other artifacts is performed.

In order to pass the exercise all correctly executed task results marked with an arrow bullet (➢) must be personally presented to a course assistant during the scheduled laboratory exercise.

Implementation

The data and toolbox files used in this exercise can be found in the Biosignal Processing II course webpage (see the Noppa system link in the footer of this document).

Download the data file '521282S_eegdata.mat' containing the EEG data matrix variable 'signal' (with the EOG channels), channel labels 'channelnames', time vector 't', spread matrix 'spread' (for easier plotting of the data), and sampling frequency 'Fs'. Also, download the FastICA toolbox '521282S_fastica.zip' and extract the contents into a suitable working folder (remember to add the toolbox into matlab path).

The matlab DSP toolbox (see 'help dsp') is used to design and implement the basic filters and the LMS adaptive filter. Familiarize yourself with the 'fdatool' filter design tool and the 'dsp.LMSfilter' function. Also, study the contents of the FastICA independent component analysis (ICA) toolbox, especially the 'fastica' function.

(NOTE! The FastICA graphical user interface 'fasticag' might be buggy and not be able to load data on the current Matlab version => give your data as a command line argument instead, e.g. 'fasticag(data_variable)')

Store your solutions in a single script code m-file that provides the required task results.

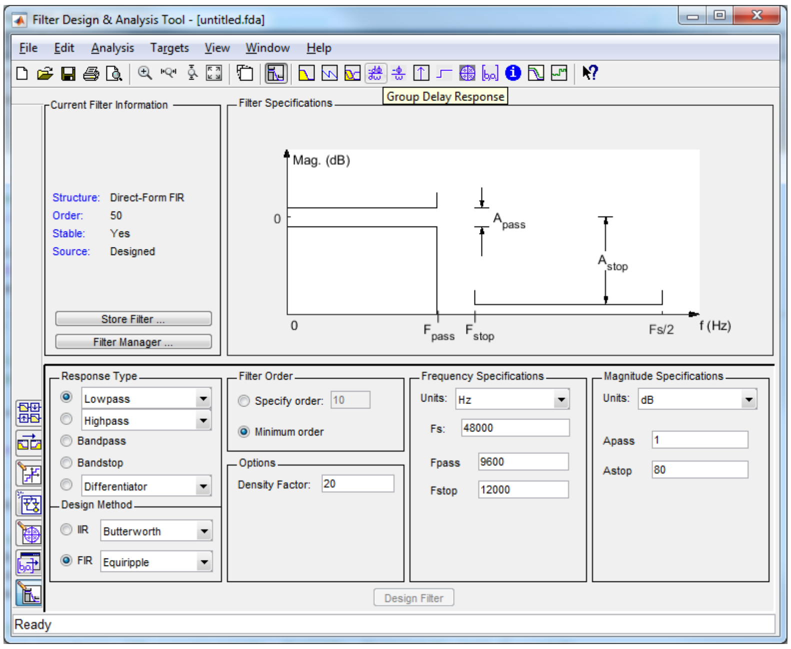
1. Plot the Raw EEG data.

Load the data file ‘521282S_eegdata.mat’. Plot the raw EEG data ‘signal’ using the time vector ‘t’ into a separate figure (tip: you can use the ‘spread’ matrix by adding it to the ‘signal’ to separate the data when drawing into a single plot).

- ➢ Show the plotted raw EEG data and observe the different artifacts/noises.
  - o NOTE! Remember to use the editor to write all of your commands into a single script file

2. Basic filtering.

Launch the filter design tool ‘fdatool’.



Current Filter Information
Structure: Direct-Form FIR
Order: 50
Stable: Yes
Source: Designed

Filter Specifications

Response Type: [✓ Lowpass] Highpass Bandpass Bandstop Differentiator

Design Method: IIR Butterworth [✓ FIR Equiripple]

Filter Order: Specify order: 10 [✓ Minimum order]

Options: Density Factor: 20

Frequency Specifications: Units: Hz; Fs: 48000; Fpass: 9600; Fstop: 12000

Magnitude Specifications: Units: dB; Apass: 1; Astop: 80

Ready

2.1. Filter design

Design the following filters, export the codes to files (File->Generate MATLAB Code->Filter Design Function), and add the filters to your working script:

- Low pass (LP) filter for high frequency noise removal
  - o Response Type: Lowpass, Design Method: FIR (Window), Filter Order: 200, Options: Scale Passband=Yes; Window=Hamming, Frequency Specifications: Fs=200Hz, Fc=90Hz
- Notch filter for PLI removal
  - o Response Type: Notching, Design Method: IIR (Single Notch), Frequency Specifications: Fs=200Hz; Fnotch=50Hz; Q=35, Magnitude Specifications: Apass=3dB
- High pass (HP) filter for trend removal

The material used in this exercise can be found in the course web page at https://noppa.oulu.fi/noppa/kurssi/521282s/harjoitukset

  - Response Type: Highpass, Design Method: FIR (Least-Squares), Filter Order: 500, Frequency Specifications: Fs=200Hz; Fstop=0.1Hz; Fpass=0.5Hz, Magnitude Specifications: Wstop=1; Wpass=1

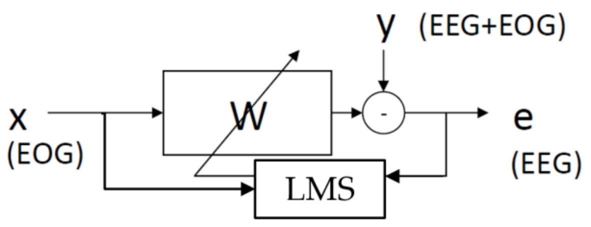
2.2. Filtering the EEG signal

Use the previously designed filters to filter for noise, PLI, and trends:

- Use the ‘filter’ command to apply the FIR LP filter to the raw EEG data ‘signal’
- Use the ‘filtfilt’ command to apply the IIR notch filter as a forwards & backwards filter for the previously LP filtered signal
  - NOTE!, ‘filtfilt’ accepts only the filter coefficients a and b
- Use the ‘filter’ command to apply the FIR HP filter for the previously notch filtered signal
- ➢ Store and plot the final LP, notch, and HP filtered EEG signal using the time vector ‘t’
  - tip: remember the ‘spread’ matrix for easy plotting

3. Adaptive Filtering

$$e_{LMS,k} = y_k - \sum_{i=0}^{n-1} w_k(i) x_{k-i}$$



$$w_{i(k+1)} = w_{ik} + 2\mu e_{LMS,k} x_{k-i}$$

Generate an LMS adaptive filter object using the ‘dsp.LMSfilter’ function with the following specifications:

- Length=11, Method = ‘LMS’, StepSize=0.6, WeightsOutputPort=’false’

Use the generated LMS filter object to filter each previously LP, notch, and HP filtered EEG data channel individually using the EOG1 and EOG2 channels (see ‘channelnames’ for corresponding channel). (NOTE! Do not remove the EOG channels from the EEG data matrix but let them be in both of the inputs):

- Run the LMS filter first using the previously filtered EEG data and the first EOG1 channel as inputs, run 10 iterations to stabilize the filter (use the ‘step’ command to run the filter one iteration, e.g. [out1, out2] = step(LMSfilter, eog_channel_in, eeg_signal_matrix_in))
  - Tip: make a ‘for’ loop for the channels and run the LMS filter 10 times inside, then get the 10th iteration outputs for each channel

The material used in this exercise can be found in the course web page at https://noppa.oulu.fi/noppa/kurssi/521282s/harjoitukset

- ➢ Store and plot the EOG1 only LMS filtered EEG signals (i.e. the second output) using the time vector ‘t’
    - NOTE: The corresponding EOG signal should now be flat

- Run the LMS filter a second time but now using the previously EOG1 channel LMS filtered EEG data and the second EOG2 channel (i.e. the previously EOG1 LMS filtered EOG2 channel) as inputs (run ‘step’ 10 iterations)
- ➢ Store and plot the now EOG1 and EOG2 LMS filtered EEG signals using the time vector ‘t’
    - NOTE: Both EOG signal channels should now be flat

4. BSS artifact removal with Independent Component Analysis (ICA)

Use the FastICA toolbox command ‘fastica’ to generate ICA decomposition for the previously LP, notch, and HP filtered EEG data (i.e. do not use the LMS filtered signals).

- Run ‘fastica’ using default parameters on the EEG signals and store the independent components (IC), the mixing matrix A, and the unmixing matrix W.
    - Tip: It is handy to set the random number generator first to a known state to produce deterministic results as ICA might give you the ICs in different order each time (e.g. rng(666))
- ➢ Store and plot the ICs of the EEG signals using the time vector ‘t’
    - NOTE: The ICs are scaled so you need to rescale them first (e.g. divide by 100) in order to use the provided ‘spread’ matrix in plotting
- Observe the ICs and note the components that are artifacts
    - Tip: some components might also contain filtering artifacts
- Remove the artifacts by zeroing those components from the IC matrix.
- Multiply the modified IC matrix from the left with the mixing matrix A to return the components into the original EEG signals
    - NOTE: The original scale is now restored
- ➢ Store and plot the ICA BSS artifact cleaned EEG signals using the time vector ‘t’

The material used in this exercise can be found in the course web page at
https://noppa.oulu.fi/noppa/kurssi/521282s/harjoitukset